An Algorithm to Estimate Mean Traffic Speed[1] using Un-Calibrated Cameras

D.J. Dailey, F.W. Cathey, S. Pumrin
Dept. of Electrical Engineering
Box 352500
University of Washington
Seattle, WA 98195

ABSTRACT

In this paper, we present a novel approach to estimate traffic speed using a sequence of images from an un-calibrated camera. We assert that exact calibration is not necessary to estimate speed. Instead, we use: (1) geometric relationships inherently available in the image, (2) some common sense assumptions that reduce the problem to a 1-D geometry, (3) frame differencing to isolate moving edges and track vehicles between frames, and (4) parameters from the distribution of vehicle lengths, to estimate speed.

KEYWORDS

Video Image Processing, Calibration, Camera, Speed Sensor, Vehicle Length Distribution

I. INTRODUCTION

Image processing techniques have been applied to traffic scenes for a variety of purposes, including: queue detection, incident detection, vehicle classification, and vehicle counting [1], [2], [3], [4], [5]. In this paper, we present a new algorithm to estimate speed using a sequence of video images from an un-calibrated camera. This work is motivated by the large number of roadside cameras installed by DOT's to observe traffic. The cameras are typically not installed in a manner that they can easily be calibrated, and they are typically used by operators who can tilt, pan, and zoom using a joystick to change the camera calibration. The combination of movable cameras and lack of calibration makes estimating speed for un-calibrated cameras a challenge. The traffic management community typically uses time-averaged speed estimates, and so our algorithm is designed to create a time-averaged speed.

Relatively few efforts have been made to measure speed using video images from un-calibrated cameras. Some preliminary research on pixel speed estimation in images appears in [4]. In previous work, few efforts were made to map pixel speed to ground truth speed. A review of the literature on speed estimation using cameras indicates that most algorithms either use reference information in the scene or create such references interactively. For example, Worrall [6] reports an interactive tool to perform camera calibration in which an operator uses parallel road marks to identify vanishing points and then places a rectangular calibration grid on the image. Further, in [7] and [8], speed measurements are made using the known physical distance between two detection windows placed on the road image by an operator. Similarly, several other authors [9], [10] suggest estimating speed by placing two detection lines, of known separation, in the image and measuring travel times between the lines. In addition, Houkes [11] suggests the selection of 4 reference points forming a rectangle and performing off-line measurements. All these methods require the operator to perform a calibration procedure before speed estimation can be undertaken.

The general motion parameters estimation from Sawhney and Ayer [12] addressed a method to represent video scenes in a compact way. This method models a complicated scene to allow for future queries about motion features in video sequences without prior knowledge of camera calibration. Model-based maximum likelihood estimation and 2D/3D motion models are used for motion quantification. In the work presented here, the scene description uses a much simpler motion model and does not require 2D/3D models but instead requires a calibration between pixel distance and real world distance traveled.

In this paper, it is assumed that we have no control over camera movements, and thus cannot directly obtain information such as camera focus, tilt, or angle. It is further assumed that the camera parameters can change with time. In the work presented here, we are monitoring congested freeways and have neither the ability nor the authority to set permanent marks on the road. Given this scenario, we believe on-line calibration is a necessary step to enable the use of the large, installed base of Traffic Management System (TMS) cameras.

We assert that exact calibration is not necessary to estimate time averaged speed. Instead, we use: (1) geometric relationships inherently available in the image, (2) some common sense assumptions (listed below) that reduce the problem to a 1-D geometry, and (3) the parameters of the distribution of vehicle lengths, to propose a novel method that extracts scale information and estimates speed.

To describe and demonstrate our speed estimation scheme, we first review the assumptions made in formulating the algorithm. We then discuss the individual steps necessary to estimate speed and present the detailed steps in the algorithm. Finally, we present some preliminary quantitative results of the algorithm.

II. UNDERLYING ASSUMPTIONS

To create an algorithm to estimate speed from video images, we make several assumptions to simplify the problem:

1. The speed of the vehicles is finite. The speed of a vehicle has both physical and legal limits.
2. The vehicle movement is smooth. There are no sudden changes of direction in the time interval (200ms) between frames in the image sequence.
3. Motion is constrained to the road plane. Tracking of vehicles in the image sequence is a one dimensional problem.
4. The scale factor (feet per pixel) varies linearly along the direction of vehicle travel. We constrain the vehicles to be moving generally toward or generally away from the camera.
5. The lengths of the vehicles in the images are realizations from a known vehicle length distribution.

With these assumptions, the vehicles are treated as though they travel in one dimension along a straight line in the image. The vehicles are tracked across these images to obtain scale factors that estimate the real-world distance represented by pixels at various locations in the image. Using a linear function to fit to the empirical scale factors, it is possible to estimate the real-world distance traveled. Combining the distance traveled with the known frame rate allows us to estimate speed. An algorithm to perform this estimation is presented in the next section.

III. ALGORITHM OPERATION

To explain the operation of the algorithm, we identify the basic tasks necessary to estimate speed from sequential un-calibrated images. The tasks necessary to obtain speed from un-calibrated images are: (1) obtain sequential images, (2) identify the moving vehicles in the sequential images, (3) track the vehicles between images, (4) dynamically estimate the scale factor in feet per pixel, and (5) estimate speed from distance traveled and the interframe delay. An overview of the methodology for steps one and two is found in Figure 1

The algorithm operates on sets of sequential images taken from DOT CCTV cameras. The images used in this work are grey scale, 320 by 240 pixels, sampled five times per-second. The resolution and sample rate are selected to provide sufficient detail in the image to identify individual vehicles and to capture sequential images rapidly enough so that individual vehicles can be tracked between images without pattern recognition techniques (e.g., the vehicles move no more than about one vehicle length between images).

The images used in our algorithm are taken from roadside CCTV cameras installed by Washington State DOT in their traffic management role. The DOT transports the video from the roadside cameras to the control center using a dedicated fiber system. In the control center, operators can pan, tilt, and zoom the cameras using a joystick. The cameras are actively being used for traffic management activities. No camera calibration information is available for these cameras, and it is the purpose of this work to demonstrate that the images from such cameras can be used as an alternative speed estimate. The video is digitized at a rate of five-frames per second and stored in files using the JPEG image format. These JPEG files are the sequential images required for step one in the outer loop of the algorithm. Sets of three sequential images are used in the inner loop of the algorithm. The left side of Figure 1 shows three example images.

Each of the images is first median filtered, using a 3x3 kernel, to remove high frequency noise in the images [13], [14] and then window-level filtered to enhance the intensity ranges and adjust the contrast and brightness in the images.

To identify the moving vehicles in the images, the non-moving background must be removed. Two basic techniques to remove the static background information appear in the literature. The first technique obtains a frame with only the background that can be subtracted from the frames in which there are vehicles [15]. This frame is then updated to match the current lighting levels [2]. This method is not only computationally expensive, but it may be impossible, on congested freeways, to obtain an image with the correct lighting level and with no vehicles present. The second technique uses sequential frames to perform forward and backward differences between the frames [16], [15], [17]. Vieren [17] suggests using interframe differences with a differential operator to extract moving edges.

The algorithm presented here uses interframe differences and then applies a Sobel edge detector (labeled Sobel in Figure 1) to the resulting image. The resulting images are thresholded to obtain binary images. The image in the upper right of Figure 2 is the binary image that results from applying the Sobel edge detector to the difference of the top two images in the left column of Figure 2. The image in the right center of Figure 2 is the difference image that results from the middle and bottom images in the left column. These two binary images (the top and middle images in Figure 2) are intersected to obtain a moving edge image appearing in the lower right of Figure 2.

Examining the lower right image in Figure 2 shows that while we have identified the moving edges, those edges do not make closed polygons identifiable as individual vehicles. To overcome this problem and create closed curves, we use two morphological operations. We enhance the moving edge image by sequentially applying dilation and erosion [18]. Dilation of an object is the translation of all of its points with regard to a structural element followed by a union operation. Dilation is applied to the binary image to close the curves in the moving edge image; it also expands the overall size of the area enclosed. Erosion is then used to shrink the object back to the original size. In the algorithm presented, a 3x3 structural element is used in steps 2f and 2g to perform dilation and erosion.

After the application of the morphological operators, the moving edges are filled in to create moving blobs. These moving blobs represent the vehicle motion in the images. Past work has asserted that the convex hull surrounding a vehicle in an image is a good approximation of the projection of a vehicle in the image [19]. To characterize the moving blobs, we calculate the convex hull and calculate the centroids for the blobs. The area of each blob is calculated and is indexed by the vertical location of its centroid in the image. This area is an indicator of blob size relative to its location in the image. In the following steps, we use only blobs whose size is greater than 7x7 pixels but less than twice the area for the typical blob in that area of the image.

Having located a vehicle in one image, the vehicle is tracked across images by enforcing co-linearity of the centroids of the convex hulls. The left side of Figure 3 presents a representation of three convex hulls with centroids $(x_1, y_1), (x_2, y_2)$, and (x_3, y_3). The vehicle is tracked as it moves along the line at an angle (α) relative to the horizontal scan lines in the image. In the work presented here, a minimum value of 0.95 of the linear regression correlation coefficient,

$$r = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{(n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2)^{1/2}(n\sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2)^{1/2}}, \qquad (1)$$

is used to identify co-linear centroids and track a vehicle. This completes tasks one through three necessary to estimate speed.

The fourth task, making an estimate of the camera calibration parameters, is common to most tracking problems using cameras [20]. Figure 4 shows the geometry of the problem. We define two coordinate systems. The first is a camera/sensor centered coordinate system as shown at the top of Figure 4 where the x_s axis is the line of sight of the camera and the y_s-z_s plane is parallel to the image plane. The second coordinate system, as shown at the bottom of 4, is an earth fixed system where the ground plane is defined by x_e and y_e with the vertical axis z_e downward. We assume that the horizontal axis of the image plane is parallel to the ground plane and that the camera's image axis orientation is at a down angle of ϕ with respect to the horizontal. To map between the sensor/camera coordinate system and the earth coordinate system, we identify an affine transformation with a rotation matrix $\mathbf{H}_{se}$ and a translation vector $\mathbf{B}$ such that,

$$\mathbf{X}_e = \mathbf{H}_{se}\,\mathbf{X}_s + \mathbf{B} \tag{2}$$

where

$$\mathbf{X}_e = \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} \quad \mathbf{X}_s = \begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} \quad \mathbf{H}_{se} = \begin{pmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ -h \end{pmatrix}, \tag{3}$$

where the $\mathbf{B}$ vector is the location of the eyepoint of the camera in earth coordinates. The earth to sensor rotation is the inverse of $\mathbf{H}_{se}$ and since $\mathbf{H}_{se}$ is orthogonal, $\mathbf{H}_{es}$ is $\mathbf{H}_{se}^T$[21].

It is useful to define a basis of unit vectors in the earth coordinate system: (1) one along the roadway $\bar{\mathbf{L}}_e$, (2) one perpendicular to the roadway $\bar{\mathbf{W}}_e$, and (3) one in the vertical direction $\bar{\mathbf{K}}_e$. These are written

$$\bar{\mathbf{L}}_e = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} \quad \bar{\mathbf{W}}_e = \begin{pmatrix} \sin\theta \\ -\cos\theta \\ 0 \end{pmatrix} \quad \bar{\mathbf{K}}_e = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{4}$$

where θ is relative to the x axis in earth coordinates. We also define a set of coordinates l, w, k along these directions. A differential change in position along the road in earth coordinates is $d\bar{\mathbf{X}}_e = \mathbf{H}_e\, dl$, where dl is the magnitude of the differential distance along the road. The same change in position in sensor coordinates is $d\bar{\mathbf{X}}_s = \mathbf{H}_{es}(\bar{\mathbf{L}}_e\, dl)$ or, explicitly

$$\begin{pmatrix} dx_s \\ dy_s \\ dz_s \end{pmatrix} = \begin{pmatrix} \cos\theta\cos\phi\, dl \\ \sin\theta\, dl \\ -\cos\theta\sin\phi\, dl \end{pmatrix}. \tag{5}$$

We define the coordinates in the image plane to be u and v and assert a simple perspective relationship [22] between the 3-D world and the image plane,

$$u = b\,\frac{y_s}{x_s} \qquad v = b\,\frac{z_s}{x_s} \tag{6}$$

where b is a scaling constant. Using these perspective relationships and differential calculus we can obtain an expression for differentials in the image plane,

$$du = \frac{1}{x_s}(b\,dy_s - u\,dx_s) \qquad dv = \frac{1}{x_s}(b\,dz_s - v\,dx_s). \tag{7}$$

The vehicles are moving on the road and so $z_e = 0$ for a vehicle and from (3) we get $h = z_s\,\cos(\phi) + x_s\,\sin(\phi)$. Using this result and the expression for v from (6), we find that x_s is

$$x_s = \frac{bh}{v\,\cos\phi + b\sin\phi}. \tag{8}$$

In this work we make the assumption that the vehicles are traveling generally toward or away from the camera. This assumption allows us to use the projection of the vehicle size/motion on the v axis as the measurement of interest. Combining equations (5), (6), (7) and (8), we construct the differential relationship between motion in the direction of the roadway and changes in the v direction in the image plane,[1]

$$\frac{\partial v}{\partial l} = -\frac{\cos\theta(v\cos\phi + b\sin\phi)^2}{bh}. \tag{9}$$

In a similar fashion, differentials for the other directions, $\bar{\mathbf{W}}_e$ and $\bar{\mathbf{K}}_e$ from (4), can be constructed,

$$\begin{aligned} \frac{\partial v}{\partial w} &= \frac{(v\cos\phi + b\sin\phi)^2\sin\theta}{bh} & (10)\\ \frac{\partial v}{\partial k} &= \frac{(v\cos\phi + b\sin\phi)(v\sin\phi - b\cos\phi)}{bh}. & (11) \end{aligned}$$

If the mean vehicle length $\Delta\bar{l}$, mean vehicle width $\Delta\bar{w}$, and mean vehicle height $\Delta\bar{k}$, as taken from [23], are used with these differentials, the projection of a vehicle on the roadway in the v direction in the image plane can be written,

$$\Delta v \approx \Delta\bar{l}\,\frac{\partial v}{\partial l} + \Delta\bar{k}\,\frac{\partial v}{\partial k} + \Delta\bar{w}\,\frac{\partial v}{\partial w}. \tag{12}$$

The mean height and width can be expressed as a fraction of the mean length, $\Delta\bar{w} = a_w\,\Delta\bar{l}$ and $\Delta\bar{k} = a_k\,\Delta\bar{l}$.

$$\frac{\Delta v}{\Delta\bar{l}} \approx \frac{\partial v}{\partial l} + \left\{ a_k\,\frac{\partial v}{\partial k} + a_w\,\frac{dv}{dw} \right\}. \tag{13}$$

Using this with equations (9), (10), and (11), we get a quadratic in v,

$$\begin{aligned} \frac{\Delta v}{\Delta\bar{l}} = {} & v^2 \left[\frac{\cos\phi(a_k\sin\phi + \cos\phi(a_l\sin\theta - \cos\theta))}{bh}\right] \\ - {} & v \left[\frac{a_k\cos 2\phi + \sin 2\phi(a_w\sin\theta - \cos\theta)}{h}\right] \\ - {} & \left[\frac{b\sin\phi(a_k\cos\phi + \sin\phi(\cos\theta - a_w\sin\theta))}{h}\right], \end{aligned} \tag{14}$$

that depends on b, h, θ and ϕ which are assumed constant but are not observable. We define the inverse of the scaling function we use to map displacements along the roadway to pixels in the v direction as

$$\frac{1}{q(v)} = \frac{\partial v}{\partial l} \approx \frac{\Delta v}{\Delta\bar{l}} + \beta \tag{15}$$

which has an inverse quadratic relationship with v and a bias β associated with the two rightmost terms in equation (13). The cameras in use are not calibrated but a parametric

[1] If the vehicles are instead moving generally across the image, a similar development in terms of $\frac{\partial u}{\partial dl}$ is possible.

study of the form of equation (13) allows us to bound our bias error. To evaluate this bias we create a function

$$\frac{\beta}{\left(\frac{\partial v}{\partial l}\right)} = \frac{\left(a_k \frac{\partial v}{\partial k} + a_w \frac{\partial v}{\partial w}\right)}{\left(\frac{\partial v}{\partial l}\right)} \tag{16}$$

that represents the fraction of the scaling factor by which β biases our estimate. Figure 5 presents representative surfaces created by this function when evaluated with values of b and h, estimated by hand and by WSDOT respectively, and varying the values of α, ϕ and v. Each of the surfaces is created at a constant value of α and u that in turn implies a value for θ

$$\theta = \arctan(\frac{u\,\cos\phi + b\,\frac{\sin\phi}{\tan\alpha}}{b}). \tag{17}$$

These surfaces indicate that the bias term can have significant effects on the results of our approximation if the measurements are not restricted to regions of the images where the approximation is acceptable.

The algorithm is constrained to use vehicles from the image that project a length of at least seven pixels in the v direction. Figure 6 shows the surface of equation (12) which is the vehicle projection in pixels along the v axis. Using representative values from equation (12) we can visualize the seven pixel constraint in the bottom plane of Figure 5. The darkened area is the combination of the variables in equation (12) that will produce an acceptably long vehicle image. Near $\phi = 0$ and above $v = 0$ the seven pixel criteria is not met, this area is shown in white in Figure 5. The seven pixel constraint assures that measurements from this region will not be used and eliminates the contribution of the very large bias in this region. In addition, the requirement for the vehicles moving toward or away from the camera restricts the acceptable α values to $50^o < \alpha \leq 90^o$. With these constraints the bias introduced is of acceptable levels when compared to the overall variability in the observed data.

We make individual estimates of the scale factor ($\hat{q}$) as the ratio of the mean vehicle length ($\Delta\bar{l}$) and the estimate of vehicle pixel length (Δv)from the image. A representative data set is shown in Figure 7. Superimposed on the data set are two functions. The line with curvature is the least square best fit of the function

$$\frac{1}{c_1 v^2 + c_2 v + c_3} \tag{18}$$

to the observed scale factor estimates. This function has the same form as equation (15) or the inverse of equation (13). The second line in Figure 7 is of the form

$$q(v) = mv + b \tag{19}$$

or a linear approximation to the observed scale factor estimates. The observed estimates have a variance larger than the difference between the two models. The largest possible contribution from the quadratic term in equation (18) is less than 10% of the value which is smaller than the variance. Based on the combination of the variability of the data, which makes the estimation of higher order terms of questionable value, and the overall trend in the data, we assert that we can approximate this function using a linear relationship.

The distance traveled is the integral of this function along the v direction,

$$d = \int_{v_1}^{v_2} q(v)dv. \tag{20}$$

Finally, having an estimate of the distance traveled, we use the interframe sample time, Δt, to estimate the vehicle speed,

$$\hat{S} = \frac{d_k}{\Delta t}. \tag{21}$$

This provides an estimate of speed from un-calibrated cameras.

IV. ALGORITHM

The algorithm operates on a series of at least five sequential images. The inner loop operates on sequential groups of three images to create one enhanced image (see Figure 1). The outer loop uses a sequence of enhanced images to estimate speed.

Outer Loop

1. Obtain five or more sequential images (320x240), gray scale at five frames per second (e.g., $(I_i, I_{i+1}, I_{i+2}, I_{i+3}, I_{i+4}, ..., I_{i+N})$ where $N \geq 5$)
2. Create sets of three sequential images (e.g., (I_i, I_{i+1}, I_{i+2}) is the ith set of $(N-2)$ sets)

   Inner Loop For each of the sets of three sequential video images, perform the following:

   (a) Median filter and window-level each image.
   (b) Difference the first and second images $(I_i - I_{i+1})$ as well as the third and second images $(I_{i+2} - I_{i+1})$ to get two difference images.
   (c) Apply a Sobel edge detector to the difference images to obtain edge images $Sobel(I_{i+2} - I_{i+1})$ and $Sobel(I_i - I_{i+1})$.
   (d) Threshold the edge images to create binary images.
   (e) Intersect the two binary images to obtain the moving edge image $(ME(I_{i+1}))$ for the I_{i+1} image:

   $$\begin{aligned} ME(I_{i+1}) = & \ Threshold(Sobel(I_{i+2} - I_{i+1})) \\ \cap & \ Threshold(Sobel(I_i - I_{i+1})). \end{aligned}$$

   (f) Apply dilation to the moving edge image.
   (g) Apply erosion to the moving edge image.
   (h) Identify the set of points $\mathbf{C}_j(I_{i+1})$ for the j convex hulls in the moving edge image $ME(I_{i+1})$.
   (i) Calculate centroid $\rho(i+1, j) = (u, v)$ for the jth convex hull in image I_{i+1}.
   (j) Calculate the set of points for the bounding boxes $\mathbf{B}_j(I_{i+1})$ for the j convex hulls.

   End of the inner loop

3. Select sets of co-linear centroids $(\rho(i+1, j), \rho(i+2, j), \rho(i+3, j), \quad ... \quad \rho(N-2, j))$ in sequential images and estimate a best fit line through these points. The slope of this line is the tangent of the angle of motion α for the jth centroid in the series of images.

4. For each of the collinear bounding boxes in sequential images, estimate the pixel length $L(i+1,j)$ along the direction v,

$$L(i+1,j) = sup(v : v \in \mathbf{B}_j(I_{i+1})) - inf(v : v \in \mathbf{B}_j(I_{i+1})),$$

where *sup* is the *supremum* or *least upper bound* and *inf* is the *infimum* or *greatest lower bound* [24].

5. Estimate the scale factor $\hat{q}_k$ (feet/pixel) for the v location of the centroid $\rho(i+1,j)$ using the mean vehicle length $\Delta\bar{l}$,

$$\hat{q}(i+1,v) = \frac{\Delta\bar{l}}{L(i+1,j)},$$

and incorporate this estimate in the collection of data used to estimate the scale factor function from equation (19). Vehicle dimensions and percentage distribution of traffic volumes from [23] are used to estimate the mean vehicle length.

End Outer Loop

6. Use the collected set of scale factor estimates to estimate the slope (m) and intersection (b) of the scale factor function $q(v|m,b)$ using

$$\min_{(m,b)} \|q(v|m,b) - \hat{q}(i+1,v)\|^2 \quad \forall i,$$

where $q(v|m,b) = mv + b$ and v is the distance along a vertical axis in the image.

7. Estimate the interframe distances,

$$d_k = \int_{v_k}^{v_{k+1}} q(v)dv \qquad \forall \quad k \in (i+1, N-2).$$

8. Estimate the mean of interframe distances $E(d_k)$ and use the ratio of the interframe mean and the frame rate (Δt) to estimate speed,

$$\hat{s} = \frac{E(d_k)}{\Delta t}.$$

V. EMPIRICAL RESULTS

Preliminary results of applying the algorithm to images from a variety of lighting conditions indicate that it is a viable alternative to calibrated cameras, when the region of the image used to estimate speed is appropriately constrained. Images from roadside cameras are used with appropriate constraints to obtain speed estimates suitable for comparison both with ”ground truth” and inductance loop speed estimates.

The speed estimates presented here are made using three practical constraints. The first constraint is that only vehicles whose projection is over 7 pixels in length are used. The second constraint is that only vehicles identified in the lower half of the image are used for scale factor and speed estimation. The third constraint is that the apparent angle of the roadway in the image be $(50 < \alpha < 130)$

The algorithm accuracy is evaluated in two ways. First, a ground truth estimate of the vehicle speed is compared to the algorithm results, and second, the time average algorithm result is compared to equivalent inductance loop speed measurements.

The ground truth speed is obtained by placing calibration lines on the highway and measuring "by hand" the time for individual vehicles to pass the sequential highway markings in sequential images. This ground truth individual vehicle speed can then be compared to the estimates derived from the algorithm.

The algorithm presented here depends upon using the mean of a distribution of vehicle dimensions. A single vehicle represents a realization from this distribution and not the mean. As a result, there will be errors in the estimate of any individual vehicle speed. If speed estimates are made for a number of vehicles, the errors in these estimates will have a distribution, and the mean of this distribution will be zero if the algorithm is performing well.

Figure 8 is a histogram of the error between the individual speed estimates and the ground truth vehicle speed for a series of 190 images over forty seconds. These results are for vehicles being tracked by the algorithm without regard to lighting effects. The distribution has zero mean, and that suggests that using this methodology with a large number of vehicles will allow for an accurate estimate of a mean traffic speed. This requirement for large numbers of vehicles to get a mean value implies that it will take 20 seconds to obtain a sampling with a suitable number of vehicles. This time frame makes the measurements from this technique analogous to the speed measured using twenty second averaged inductance loop measurements.

Traffic speeds derived from inductance loops are also compared with the algorithm estimates. The inductance loop speeds come from the Traffic Data Acquisition and Distribution (TDAD) data mine (http://www.its.washington.edu/tdad) which contains data from the Seattle metropolitan region. Figure 9 is a histogram of the difference between the results of our speed estimation algorithm and the twenty second average loop speed estimate. Figure 9 is constructed from 342 video speed estimates, each created from 80 three-image sequences. The scale factor function used for these estimates is created from 100 sequences of the algorithm captured just before the comparison sequence. The mean of the difference between the estimates is 4 miles per hour indicating a bias in one of the estimators. However, the deviation about that mean is relatively symmetric, indicating that the underlying process creating the estimates, the variability of speed on the freeway, is the same. The bias may be present in either estimator; however, no ground truth calibration like Figure 8 is available for the loop detectors.

We examined the relationship between lighting conditions and the error in the estimates, and it has become clear that the shadow effects account for the set of errors over 10% in the ground truth speed estimate. Reconciling the algorithm against lighting conditions is an ongoing effort.

This paper presents a new algorithm to estimate speed from un-calibrated, relatively low resolution cameras. Un-calibrated cameras are widely available to DOT operators and can provide a valuable, additional quantitative measure for traffic operations and traveler information.

ACKNOWLEDGMENT

This work was supported by funding from Washington State Department of Transportation and Transportation Northwest (TransNow), a USDOT University Transportation Center.

References

[1] Kilger M. Video-Based Traffic Monitoring. In *International Conference on Image Processing and its Applications*, pages 89–92, 7-9 April 1992. Maastricht, Netherlands.

[2] Fathy, M., and M.Y. Siyal. An Image Detection Technique Based on Morphological Edge Detection and Background Differencing for Real-Time Traffic Analysis. *Pattern Recognition Letters*, Vol. 16, No. 12:1321–1330, 1995.

[3] Ali, A.T., and E.L. Dagless. Computer Vision for Automatic Road Traffic Analysis. In *ICARCV 90, Proceedings of the International Conference on Automation, Robotics and Computer Vision*, pages 875–879, 19-21 September 1990.

[4] Soh, J., B.T. Chun, and M. Wang. Analysis of Road Sequences for Vehicle Counting . In *1995 IEEE International Conference on Systems, Man and Cybernetics Vol.1*, pages 679–683, 22-25 October 1995. Vancouver, British Columbia, Canada.

[5] Zifeng J. Macro and Micro Freeway Automatic Incident Detection(Aid) Methods Based on Image Processing. In *IEEE Conference on Intelligent Transportation Systems*, pages 344–349, 9-12 November 1997. Boston, Massachusetts, USA.

[6] Worrall, A.D., G.D. Sullivan, and K.D. Baker. A Simple, Intuitive Camera Calibration Tool for Natural Images . In *Proceedings of the 5th British Machine Vision Conference*, pages 781–790, 13-16 September 1994. York, UK.

[7] Dickinson, K.W., and R.C. Waterfall. Video Image Processing for Monitoring Road Traffic . In *IEE International Conference on Road Traffic Data Collection*, pages 105–109, 5-7 December 1984.

[8] Ashworth, R., D.G. Darkin, K.W. Dickinson, M.G. Hartley, C.L. Wan, , and R.C. Waterfall. Applications of Video Image Processing for Traffic Control Systems. In *Second International Conference on Road Traffic Control*, pages 119–122, 14-18 April 1985. London, UK.

[9] Takaba, S., M. Sakauchi, T. Kaneko, B. Won-Hwang, and T. Sekine. Measurement of Traffic Flow Using Real Time Processing of Moving Pictures. In *32nd IEEE Vehicular Technology Conference*, pages 488–494, 23-26 May 1982. San Diego, California, USA.

[10] Hashimoto, N., Y. Kumagai, K. Sakai, K. Sugimoto, Y. Ito, K. Sawai, and K. Nishiyama. Development of an Image-Processing Traffic Flow Measuring System. *Sumitomo Electric Technical Review*, No. 25:133–137, 1986.

[11] Houkes Z. Measurement of Speed and Time Headway of Motor Vehicles with Video Camera and Computer. In *Proceedings of the 6th IFAC/IFIP Conference on Digital Computer Applications to Process Control*, pages 231–237, 14-17 October 1980. Dusseldorf, West Germany.

[12] H.S.-Sawhney and S.-Ayer. Compact representations of videos through dominant multiple motion estimation. *PAMI*, 18(8):814–830, 1996.

[13] Pratt William K. *Digital Image Processing*. John Wiley and Sons. Inc., 2nd edition, c1991.

[14] Davies E.R. *Machine Vision: Theory, Algorithms, Practicalities*. Acedemic Press, 2nd edition, 1997.

[15] Cai, Q., A. Mitiche, and J.K. Aggarwal. Tracking Human Motion in an Indoor Environment. In *International Conference on Image Processing, Vol. 1*, pages 215–218, 23-26 October 1995. Washington, D.C., USA.

[16] Kudo, Y., T. Yamahira, T. Tsurutani, and M. Naniwada. Traffic Flow Measurement System Using Image Processing. In *33rd IEEE Vehicular Technology Conference*, pages 28–34, 25-27 May 1983. Toronto, Ontario, Canada.

[17] Vieren, C., F. Cabestaing, and J.G. Postaire. Catching Moving Objects with Snakes for Motion Tracking. *Pattern Recognition Letters*, Vol. 16, No. 7:679–685, 1995.

[18] Serra Jean Paul. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.

[19] Gil, S., R. Milanese, and T. Pun. Comparing Features for Target Tracking in Traffic Scenes. *Pattern Recognition*, Vol. 29, No. 8:1285–1296, 1996.

[20] J.R.-Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the Second European Conference on Computer Vision*, pages 237–252, Santa Margherita Liguere, Italy, May 1992.

[21] B. Jähne. *Digital Image Processing*. Springer, 4th edition, 1997.

[22] E.R. Davies. *Machin Vision*. Acedemic Press, 2nd edition, 1997.

[23] American Association of State Highway and Transportation Officials. *A policy on geometric design of highways and streets, 1994*. Washington, D.C. : American Association of State Highway and Transportation Officials, 1994.

[24] Luenberger D.G. *Optimization by Vector Space Methods*. John Wiley & Sons, Inc., 1969.

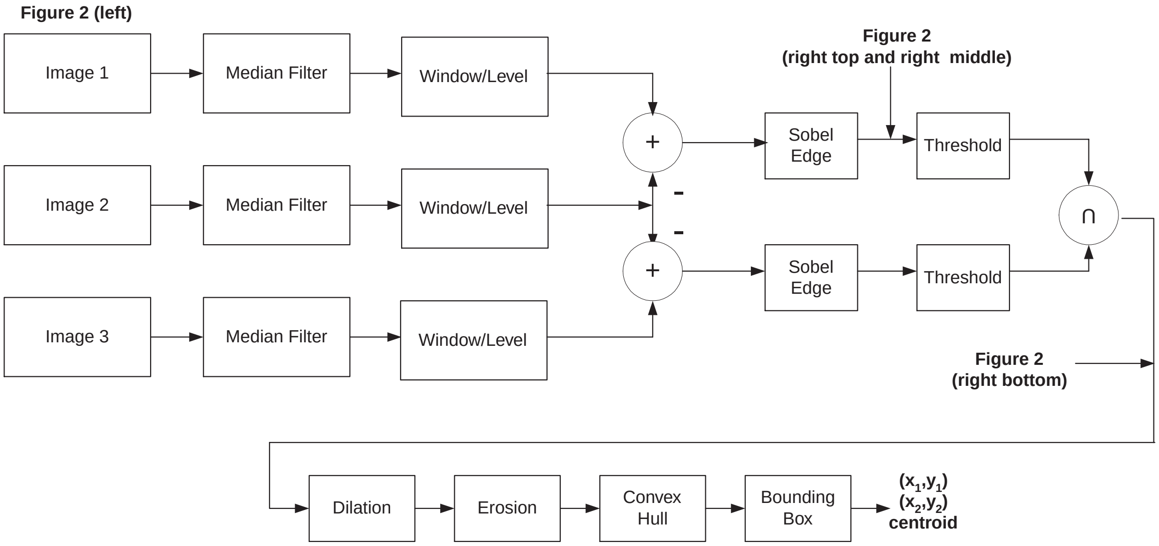

Fig. 1. **Algorithm inner loop data flow.**

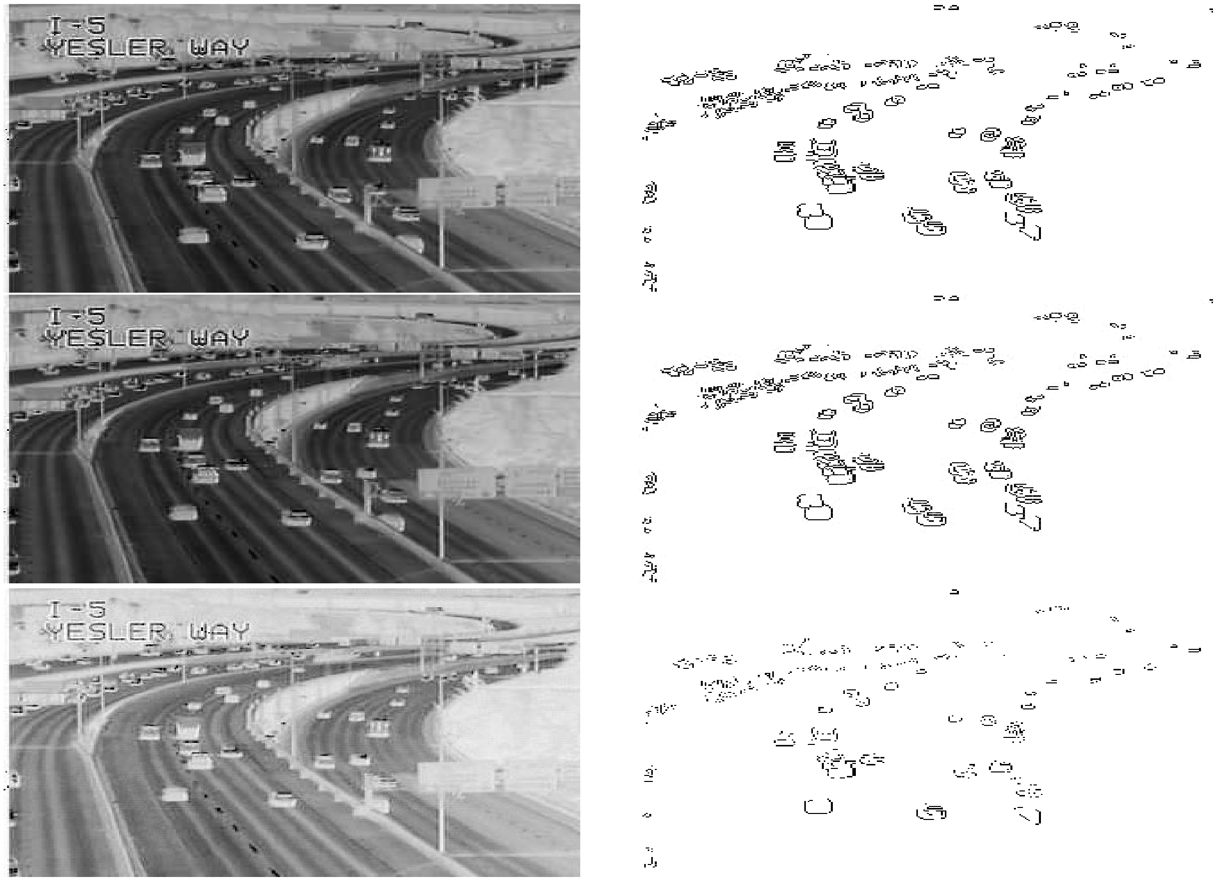

Fig. 2. **Typical image sequence (left). Sobel edges in the difference images (right top two images). Moving edge image for the middle image on the left created by intersecting the difference images.**

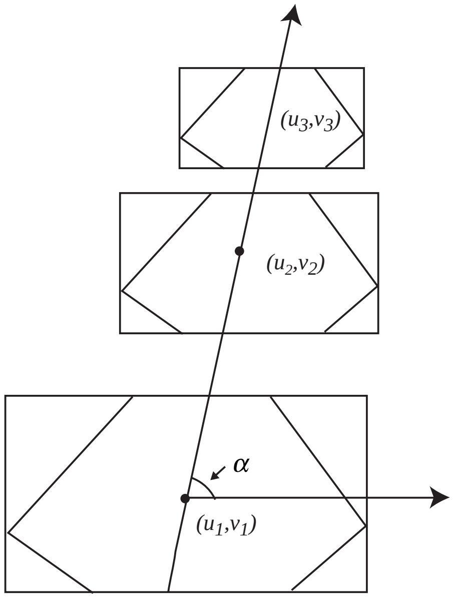

Fig. 3. **Use of centroids to establish the travel direction α.**

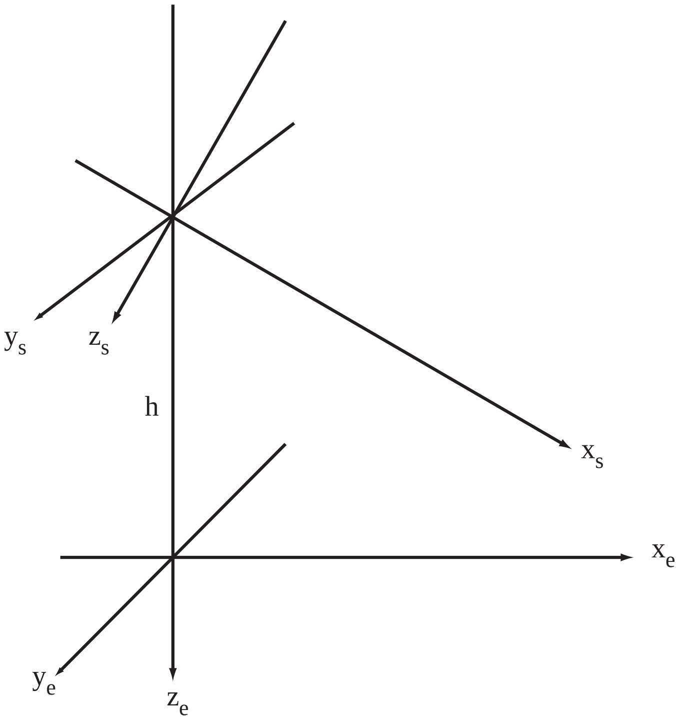

Fig. 4. **Camera geometry.**

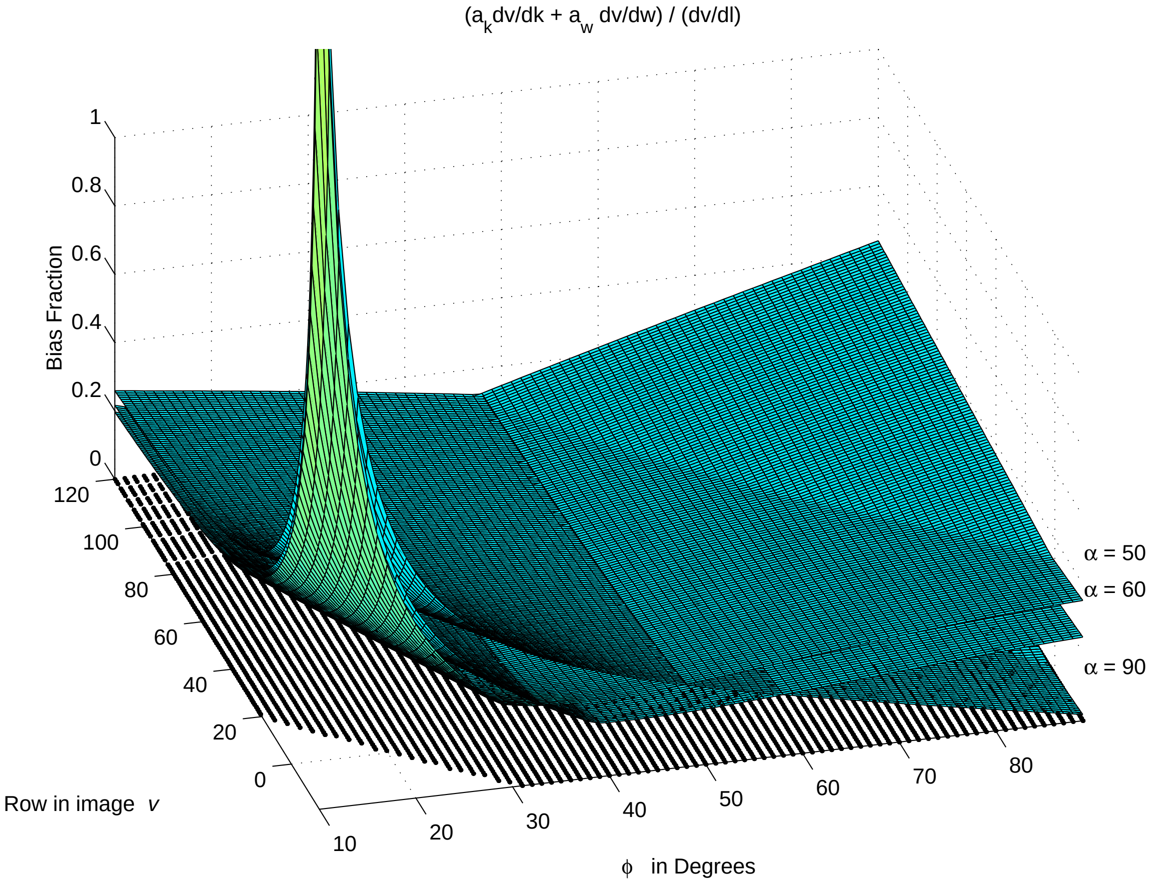

Fig. 5. **Parametric form for the bias term in equation (16) using representative parameters.**

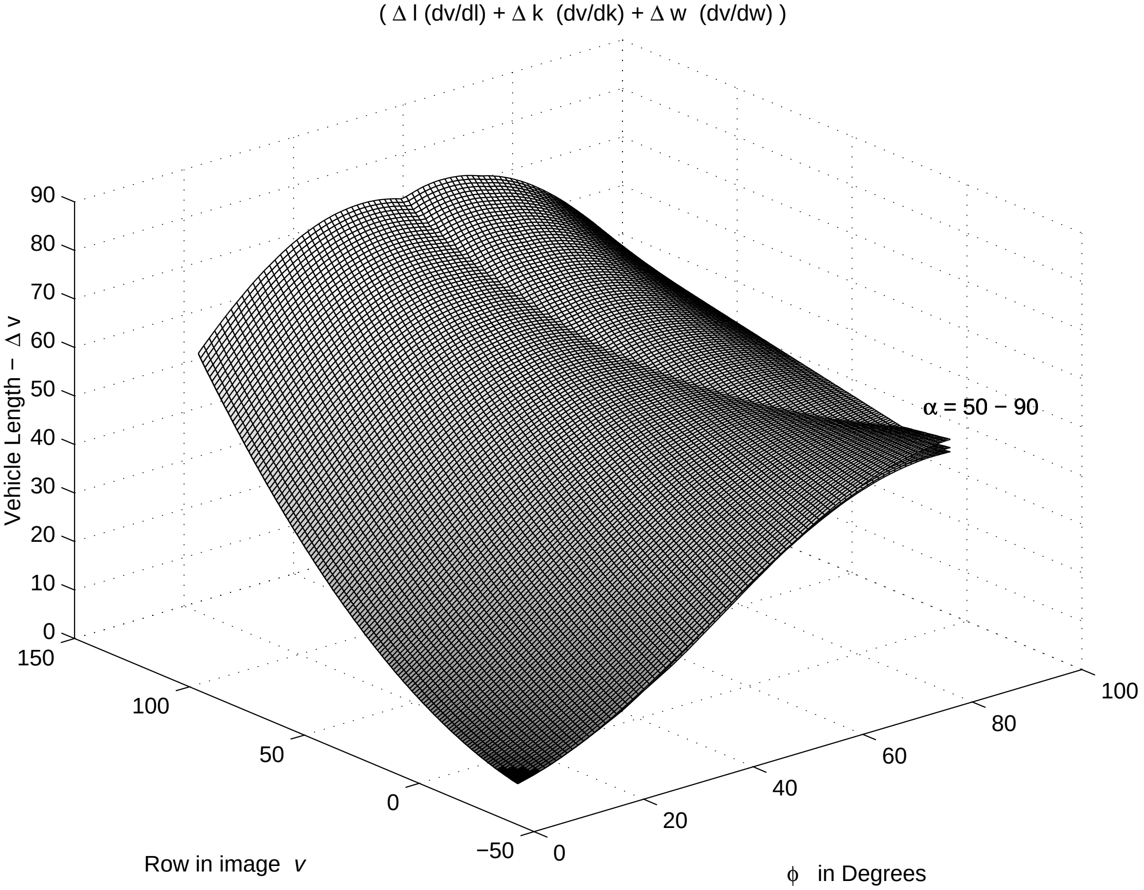

Fig. 6. **Parametric form for Δv for representative parameters.**

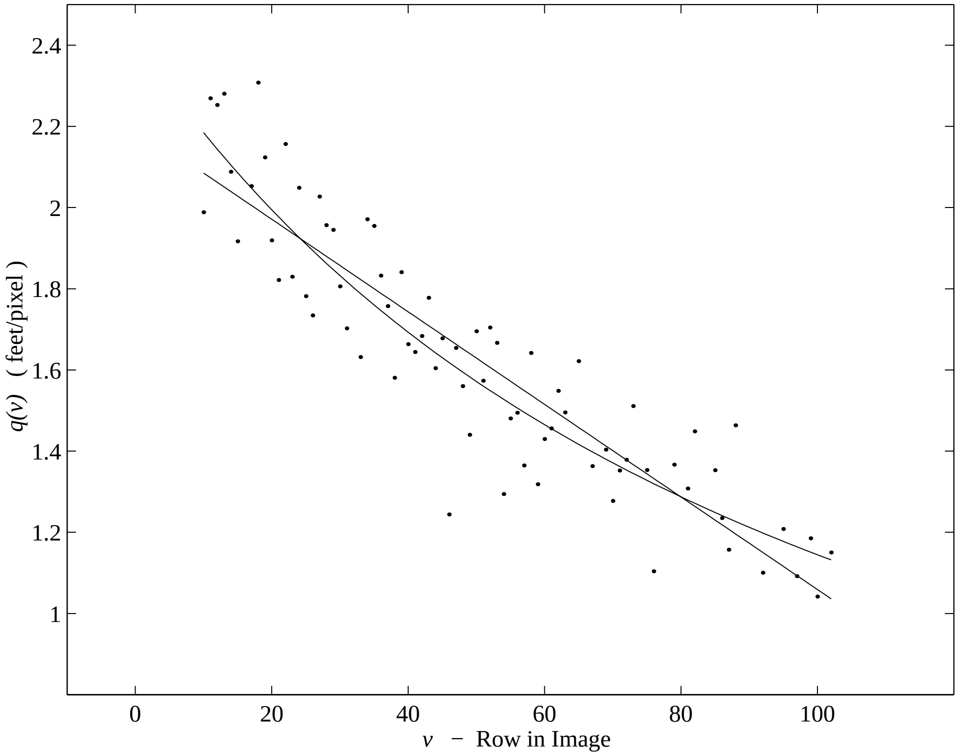

Fig. 7. **Scale factor function and observed realizations.**

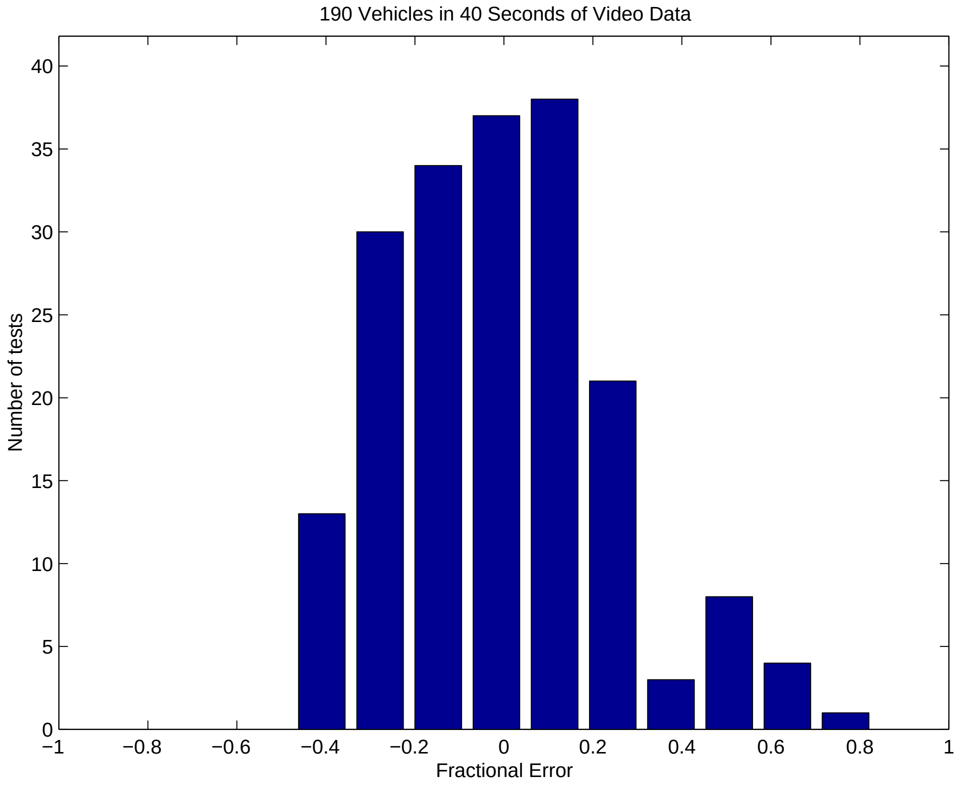

Fig. 8. **Histogram of the fraction of error for the speed estimate for single vehicles.**

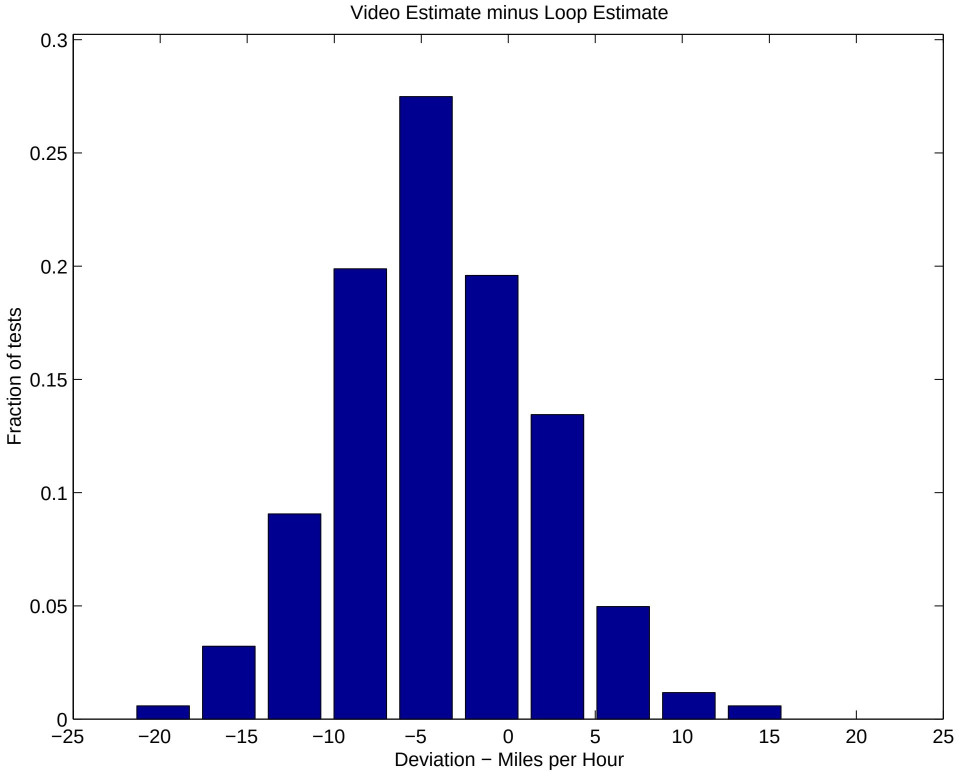

Fig. 9. **Histogram of the deviation of video speed estimates from twenty second averaged inductance loop estimates.**